Home Assignment

In this assignment, you will be building a Vue.js application that interacts with a pre-built API server written in Python.

The assignment is intended to evaluate your proficiency in the following areas: Components architecture, code writing style, data handling, UX logic, UI Aesthetics, SCSS/CSS, HTML.

The API server has already been set up and is ready for use.
The requirements for the functionality and design of the Vue.js application are outlined in the following pages.

If you have any questions about the assignment or the requirements, please don't hesitate to ask.
We want to make sure you have all the information you need to succeed in this project.

Pokédex is a pokemon exploring application.

There are two pages. **Pokemons inventory** (main page) and **Single pokemon page**.

UI Functionality

General (should be on all pages)

Header zone

Logo (can be plain text) that says **Pokédex** on the left side.
Click on the logo should redirect to the main page.
Display the current time & day of the week on the right side. The time should be constantly updated.

Pokemons inventory page

Left sidebar zone

Search box filter

Should be placed on the top.
While typing, only the pokemons whose names include the string should be visible.
Each pokemon is displayed in a card (more details below). The text entered in the search box should be displayed in bold within the name of the pokemon on the card. For example, if the text in the search box is "sau" and the pokemon is Bulbasaur, the name displayed on the card should be "Bulba**sau**r."
Filtering is in the front. No server interaction is required.

Types filter

Should be placed below the search box.
Displays a list of all the types, as checkboxes.
If there's a current search box filter, the types that should be displayed are the ones of the filtered pokemons.
Selecting a checkbox should filter the pokemons by the selected type.
Selecting multiple checkboxes filters the pokemons by an OR operator.
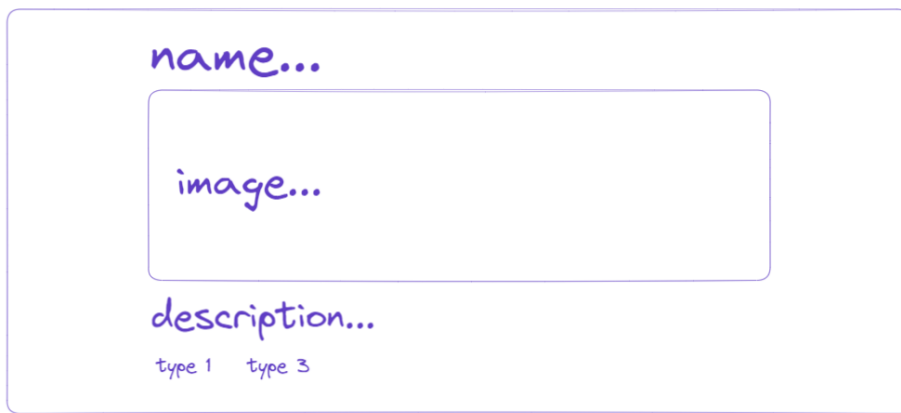
Main content zone

Each pokemon should be displayed as a card.
Each card contains a name, image, description (of max 2 lines), and the types of the pokemon.
The Card should be lazy loaded. That means, render the card only when it should be visible. Infinite scroll style.
Click on the card will redirect to the pokemon's page.

Single pokemon page

name...

image...

description...

type 1   type 3

Display the name, image, full description, and types.

Bonus

name...

image...

description...

type 1   type 3

evolution
1
(current)   →   evolution
2

Display all the evolutions of the pokemon. **All** of the past, and **all** of the future ones.
Click on evolution will redirect to the evolution's page.
The current pokemon in the evolutions should be bold and not clickable.

Notes

- Use Vue.js framework.
- The application should be responsive and supported in mobile.
- Use any external libraries you want. There are a lot of open-sources for inputs, lazy loading, etc...

God is in the details.

Loaders, error handling, lazy loading, use of open-sources etc.. are super important.

Read carefully the instructions.

Good luck! 😁